

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Atty. Docket No.: 2551-026

Douglas M. BLAIR

Appln. No.: 09/881,234

Group Art Unit: 1631

Filing Date: June 14, 2001

Examiner: Smith, C.

For: APPARATUS AND METHOD FOR PROVIDING SEQUENCE DATABASE
COMPARISON

* * * * *

APPEAL BRIEF TRANSMITTAL

* * * * *

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Enclosed, please find:

1. Appellants' Brief on Appeal (Original with 2 copies);
2. Petition for Extension of Time; and
3. Check in the amount of \$220.00 to cover the fees for the Appeal Brief (\$165.00) and the Extension of Time (\$55.00).

Respectfully submitted,

03/12/2004 BABRAHA1 00000108 09881234

01 FC:2402

165.00 OP

03/12/2004 BABRAHA1 00000108 09881234

02 FC:2251

55.00 OP

Christy B. Kil

Christopher B. Kilner

Registration No. 45,381

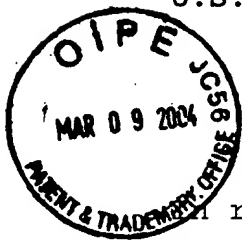
Roberts Abokhair & Mardula, LLC

11800 Sunrise Valley Drive

Suite 1000

Reston, VA 20191

(703) 391-2900



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Re Application of

Atty. Docket No.: 2551-026

Douglas M. BLAIR

Appln. No.: 09/881,234

Group Art Unit: 1631

Filing Date: June 14, 2001

Examiner: Smith, C.

For: **APPARATUS AND METHOD FOR PROVIDING SEQUENCE DATABASE
COMPARISON**

* * * * *
APPELLANT'S BRIEF ON APPEAL UNDER 37 C.F.R. § 1.192
* * * * *

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

In accordance with the provisions of 37 C.F.R. § 1.192,
Appellant submits the following:

I. REAL PARTY IN INTEREST

Based on information supplied by Appellants, and to the best
of Appellants' legal representatives' knowledge, the real party
in interest is the assignee, Parabon Computation, Inc.

II. RELATED APPEALS AND INTERFERENCES

Appellants, as well as Appellants' assigns and legal
representatives are unaware of any appeals or interferences which

will be directly affected by, or which will directly affect, or have a bearing on the Board's decision in the pending appeal.

III. STATUS OF CLAIMS

Claims 1-23 are currently pending. No claims have been allowed. No claims have been canceled. Claims 1-23 are appealed. Claims 1-23, as amended herewith, are set forth in the attached Appendix.

IV. STATUS OF AMENDMENTS

An amendment has been filed herewith to eliminate the alleged indefiniteness of the abbreviations CPU, ID, and BLAST from the claims so as to place the claims in better condition for appeal. The only other amendments in the application, filed November 22, 2002 and July 14, 2003 to amend the specification, were entered.

V. SUMMARY OF THE INVENTION

Appellants' disclosed and claimed invention is directed to a method and system of comparing a query and a subject database using a distributed computing platform. The databases are divided into data elements having a size within a specified range. All data elements and task definitions are sent to a master CPU of a master-slave distributed computing platform, wherein task

definitions comprise at least one comparison parameter, at least one executable comparison element, and a query and a subject data element ID/descriptor. Data elements are sent alternately from query and subject data elements. A task definition is sent for each task from the master CPU to one of a plurality of slave CPUs when all parts of a task definition and data elements referenced by the task definition are available at the master CPU. Data elements are then sent to the slave CPUs for performance of the tasks. Task results for each task are returned to a CPU.

In one of its broadest embodiments, the claimed invention is drawn to (claim 1) a method of comparing a query dataset N with a subject dataset M, comprising: dividing said query dataset N into n_N data elements having a size within a specified range and dividing said subject dataset M into n_M data elements having a size within said specified range (see figs. 1B, 3, and box 620 of figure 6; pars. [68] and [87] of the specification); determining a number of tasks for an entire comparison of datasets N and M as $n_N \times n_M$ (see box 628 of fig. 6; pars. [69] and [90] of the specification); sending all data elements and task definitions to a master CPU of a master-slave distributed computing platform, wherein task definitions comprise at least one comparison parameter, at least one executable element capable of performing comparisons, a query data element ID/descriptor, and a subject data element ID/descriptor, and wherein data elements are sent

alternately from query and subject data elements (see computer topology of fig. 5 and boxes 630-640 of fig. 6; pars. [70] and [90]-[92] of the specification); sending a task definition for each task from the master CPU to one of a plurality of slave CPUs when all parts of a task definition and data elements referenced by said task definition are available at said master CPU and sending data elements referenced by said task definition to said slave CPU (see box 650 of fig. 6; pars. [71] and [92] of the specification); performing each task on a slave CPU (see box 650 of fig. 6; pars. [71] and [92] of the specification); and returning task results for each task to said master CPU (see box 730 of fig. 7; pars. [72] and [96] of the specification).

In another of its broadest embodiments, the invention is drawn to a system (Claim 13) for comparing a query dataset N with a subject dataset M, comprising: a master CPU of a master-slave distributed computing platform; a plurality of slave CPUs capable of communication with said master CPU; and a client CPU (see fig. 5 and par. [63] of the specification) with instructions for: dividing said query dataset N into n_N data elements having a size within a specified range and dividing said subject dataset M into n_M data elements having a size within said specified range (see figs. 1B, 3, and box 620 of figure 6; pars. [68] and [87] of the specification); determining a number of tasks for an entire comparison of datasets N and M as $n_N \times n_M$ (see box 628 of fig. 6;

pars. [69] and [90] of the specification); sending all data elements and task definitions to said master CPU of a master-slave distributed computing platform, wherein task definitions comprise at least one comparison parameter, at least one executable element capable of performing comparisons, a query data element ID/descriptor, and a subject data element ID/descriptor, and wherein data elements are sent alternately from query and subject data elements (see computer topology of fig. 5 and boxes 630-640 of fig. 6; pars. [70] and [90]-[92] of the specification); said master CPU comprising instructions for: sending a task definition for each task to one of said plurality of slave CPUs when all parts of a task definition and data elements referenced by said task definition are available at said master CPU; and sending data elements referenced by said task definition to said slave CPU (see box 650 of fig. 6; pars. [71] and [92] of the specification); and said slave CPUs including instructions for: performing each task (see box 650 of fig. 6; pars. [71] and [92] of the specification); and returning task results for each task to said master CPU (see box 730 of fig. 7; pars. [72] and [93]-[95] of the specification).

The method of claim 1 and the system of claim 13 can be further limited by steps and means: i) for randomizing sequence order of each dataset if either dataset contains related sequences in a contiguous arrangement (claims 2 and 14, box 600 of fig. 6, and pars. [66] and [87] of the specification); ii) for

formatting said datasets so as to use exactly the same ambiguity substitutions (claims 3 and 15, box 610 of fig. 6, and pars. [67] and [87] of the specification); iii) wherein dividing said datasets into data elements further comprises: stripping all metadata from data; packing said data into an efficient structure; creating an index for said data and packing said index and said data in an uncompressed data structure; and compressing said uncompressed data structure into a data element using a redundancy reduction data compression method (claims 4 and 16, boxes 621-626 of fig. 6, and paragraph [68] of the specification); iv) for sending remaining data elements from a more numerous of said datasets to said master CPU followed by all task definitions for otherwise complete tasks if there are fewer data elements from one dataset (claims 5 and 17, par. [70]); and v) wherein said datasets are selected from the group consisting of genomic and proteomic databases (claims 12 and 23, fig. 2C, pars. [9] and [62] in the specification).

The broad method and system claims can also be further limited, wherein performing a task on said slave CPU further comprises: uncompressing and unpacking data from said query and subject data elements; looping through query sequences from said query data element to perform setup, preprocessing and table generation for each row of comparisons; looping through subject sequences from said subject data element and, for each pair of query and subject sequences, performing a comparison using said

executable element and finding results based on said at least one comparison parameter; and storing minimal information that will allow reconstruction of said result (claims 6 and 18, fig. 7, par. [72] of the specification), and in which storing said minimal information can optionally comprise: storing index information for said query and said subject sequence; storing bounds information for start and stop of said query and subject sub sequences; storing data that quantify fulfillment of significance criteria for a significant match; and storing an efficiently encoded representation of alignment between said bounds corresponding to a high-scoring segment pair (claims 7 and 19, figs. 4A and 4B, pars. [72], [85, and [86] of the specification), or it can further comprise storing a seed point and sum-set membership for each alignment for BLAST (claim 8, par. [72]), or it can further comprise storing task results in a task result file, said file including query and subject sequence data and metadata corresponding to the task that the results came from, metadata for the subject sequence, the partial subject sequence data corresponding to the subject bounds of the significant alignment result, and any other results data for each result in the task results (claims 9 and 20, par. [74]).

The method and system of claims 9 and 20 can further comprise generating a BLAST report for each query data element (claims 10 and 21, box 690 of fig. 6, pars. [74] and [99] of the specification), which can further comprise concatenating results

from all BLAST reports to produce a text file identical to a blastall run of said query and subject datasets (claims 11 and 21, box 690 of fig. 6, pars. [74] and [99] of the specification.

VI. ISSUES

The issues on Appeal are:

Are claims 4 and 16 indefinite under the second paragraph of 35 U.S.C. § 112 due to the use of the term "efficient structure" in the claims?

Are claims 7 and 19 indefinite under the second paragraph of 35 U.S.C. § 112 due to the use of the term "efficiently encoded representation of alignment" in the claims?

Are claims 1, 4, 6-7, 9-10, 12-13, 16, 18-21 and 23 unpatentable over the publication to Altschul et al. (1990) in view of each of U.S. Patent No. 5,706,498 to Fujimiya et al., the publication to Anderson et al. (1998), U.S. Patent No. 6,303,297 to Lincoln et al., and the publication to Matsumoto et al. (2000) as being obvious?

VII. GROUPING OF CLAIMS

Appealed claims 4 and 16 stand or fall together for purposes of the rejection of these claims under 35 U.S.C. § 112.

Appealed claims 7 and 19 stand or fall together for purposes of the rejection of these claims under 35 U.S.C. § 112.

Appealed claims 1, 4, 6-7, 9-10, 12-13, 16, 18-21 and 23 stand or fall separately based upon their individual claim limitations for purposes of the rejection of these claims under 35 U.S.C. § 103.

VIII. ARGUMENTS

Claim Rejections - 35 USC §112

In the Final Rejection, claims 1, 5, 6, 13, 17, and 18 were rejected as indefinite under the second paragraph of 35 U.S.C. § 112 due to the use of the abbreviation "CPU" in the claims; claims 1 and 13 were rejected as indefinite under the second paragraph of 35 U.S.C. § 112 due to the use of the abbreviation "ID" in the claims; claims 8, 10, 11, 21, and 22 were rejected as indefinite under the second paragraph of 35 U.S.C. § 112 due to the use of the abbreviation "BLAST" in the claims; claims 4 and 16 were rejected as indefinite under the second paragraph of 35 U.S.C. § 112 due to the use of the term "efficient structure" in the claims; and claims 7 and 19 were rejected as indefinite under the second paragraph of 35 U.S.C. § 112 due to the use of the term "efficiently encoded representation of alignment" in the claims.

Appellants traverse these rejections and submit that the claims, both as originally filed and as amended herewith, are definite.

APPELLANT'S BRIEF ON APPEAL
U.S. Application No. 09/881,234

Claims 1, 5, 6, 8, 10, 11, 13, 17, 18, 21, and 22

As previously submitted and cited in M.P.E.P. §2173.01, Appellants submit that a fundamental principle contained in the second paragraph of 35 U.S.C. § 112 is that Appellants are their own lexicographers. Appellants can define in the claims what they regard as their invention essentially in whatever terms they choose so long as the terms are not used in ways that are contrary to accepted meanings in the art. Appellants may use functional language, alternative expressions, negative limitations, or any style of expression or format of claim which makes clear the boundaries of the subject matter for which protection is sought. As noted by the court in *In re Swinehart*, 439 F.2d 210, 160 USPQ 226 (CCPA 1971), a claim may not be rejected solely because of the type of language used to define the subject matter for which patent protection is sought.

Appellants again submit that the proper focus during examination of claims for compliance with the requirement for definiteness of 35 U.S.C. §112, second paragraph as defined in M.P.E.P. §2173.02 is whether the claim meets the threshold requirements of clarity and precision, not whether more suitable language or modes of expression are available. When the Examiner is satisfied that patentable subject matter is disclosed, and it is apparent to the examiner that the claims are directed to such

patentable subject matter, he or she should allow claims which define the patentable subject matter with a reasonable degree of particularity and distinctness. *Some latitude in the manner of expression and the aptness of terms should be permitted even though the claim language is not as precise as the examiner might desire.* Examiners are encouraged to suggest claim language to applicants to improve the clarity or precision of the language used, *but should not reject claims or insist on their own preferences if other modes of expression selected by applicants satisfy the statutory requirement.*

The essential inquiry pertaining to this requirement is whether the claims set out and circumscribe a particular subject matter with a reasonable degree of clarity and particularity. Definiteness of claim language must be analyzed, not in a vacuum, but in light of:

- (A) The content of the particular application disclosure;
- (B) The teachings of the prior art; and
- (C) The claim interpretation that would be given by one possessing the ordinary level of skill in the pertinent art at the time the invention was made.

Claims 1, 5, 6, 8, 10, 11, 13, 17, 18, 21, and 22

In the present case, the original application disclosure discussed CPUs, IDs, and BLAST, and did not disclose the Examiner-suggested full-length terms "central processing units,"

"identifications," and "Basic Local Alignment Search Tool."

However, despite this lack of disclosure to the full-length terms, the Examiner knew exactly what they referred to, such that Appellants submit that the abbreviations used defined the patentable subject matter with a reasonable degree of particularity and distinctness.

However, despite the fact that the Examiner's stated reason for considering abbreviations indefinite, that it is "a standard rejection used whenever an abbreviation is placed into a claim," is both unreasonable and has no support whatsoever in the M.P.E.P., Appellants' have herein amended the specification and claims so as to include the full-length terms as an antecedent for each abbreviation in the claims to materially reduce the issues on appeal.

In view of the arguments above and the present amendments, Appellants respectfully submit that claims 1, 5, 6, 8, 10, 11, 13, 17, 18, 21, and 22 are definite.

Claims 4 and 16

With respect to the term "efficient structure," Appellants have previously directed attention to paragraph 68, wherein the term is defined: "efficient structure, e.g. 2 bits per nucleotide with appropriate encoding, 5 bits per amino acid residue with appropriate encoding, etc."

Although, as defined in the specification, the "efficient

APPELLANT'S BRIEF ON APPEAL
U.S. Application No. 09/881,234

structure" can take various forms based upon the type of molecule encoded, it does not make the term indefinite, but rather it makes the term broad. As stated in M.P.E.P. § 2173.04, breadth of a claim is not to be equated with indefiniteness. *In re Miller*, 441 F.2d 689, 169 USPQ 597 (CCPA 1971). If the scope of the subject matter embraced by the claims is clear, and if applicants have not otherwise indicated that they intend the invention to be of a scope different from that defined in the claims, then the claims comply with 35 U.S.C. 112, second paragraph.

In the present case, an "efficient structure," as illustrated by the examples, is a structure having the least bits needed to encode the possible data. In the case of nucleotides, with raw data having four possibilities -- Adenine, Cytosine, Guanine, and Tyrosine -- a 2-bit data structure is the efficient one. Likewise, one of ordinary skill in the art would recognize that there are twenty possible amino acid residues, leading to a 5-bit structure that is the efficient one for encoding 17-32 possibilities.

The presently claimed language is similar to claiming a type of compression, but relies on a choice of the data structure that minimizes the bits needed to encode the possibilities rather than the well-known and art-recognized data compression that is claimed in later portions of the claims. Although broad in scope since it depends on the data that is being encoded, an "efficient structure" is nonetheless clear and is as accurate as the subject

matter permits, just as a claim limitation specifying that a certain part of a pediatric wheelchair be "so dimensioned as to be insertable through the space between the doorframe of an automobile and one of the seats" was held to be definite in *Orthokinetics, Inc. v. Safety Travel Chairs, Inc.*, 806 F.2d 1565, 1 USPQ2d 1081 (Fed. Cir. 1986).

Indeed, the publication to Altschul et al. discusses the well-known "packing" of "4 nucleotides into a single byte." Since a byte has 8 bits and 4 nucleotides into 8 bits equals 2 bits per nucleotide, Appellants submit that one of skill in the art would clearly understand the claim language as illuminated by the specification.

Claims 7 and 19

With respect to the term "efficiently encoded representation of alignment," Appellants submit that the term has been taken out of its full context and that one of ordinary skill in the art of bioinformatics would clearly understand the meaning of the entire term, "an efficiently encoded representation of alignment between said bounds corresponding to a high-scoring segment pair." As discussed above, efficient encoding entails use of the minimum number of bits needed to represent the data and, as previously submitted to the Examiner, BLAST uses a specific data format to represent alignment pairs, such that the term is reasonably clear to one of ordinary skill in the art.

Claim 8

With respect to the term "seed point and sum-set membership," Appellants submit that one of skill in the art readily understands the meaning of the term "seed" as a matching word/string and "sum" as a part of the scoring when using BLAST such that the claimed term is sufficiently clear to one of skill in the art.

In view of the above-cited reasons, Appellants submit that claims 1-23 are definite and respectfully request reconsideration and withdrawal of the rejections. Appellants further note that claims 2-3, 5, 8, 11, 14-15, 17, and 22 have not been rejected in view of the prior art and are thus admittedly allowable upon being found definite.

Claim Rejections - 35 USC §103

Claims 1, 4, 6-7, 9-10, 12-13, 16, 18-21 and 23 were rejected under 35 U.S.C. 103 as being obvious over the publication to Altschul et al. (1990) in view of each of U.S. Patent No. 5,706,498 to Fujimiya et al., the publication to Anderson et al. (1998), U.S. Patent No. 6,303,297 to Lincoln et al., and the publication to Matsumoto et al. (2000).

To establish a *prima facie* case of obviousness, three basic criteria must be met (See M.P.E.P. Section 2143). First, there must be some suggestion or motivation, either in the references

APPELLANT'S BRIEF ON APPEAL
U.S. Application No. 09/881,234

themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988); *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992).

Second, there must be a reasonable expectation of success. This requirement is primarily concerned with less predictable arts, such as the chemical arts.

Finally, the prior art must teach or suggest each and every limitation of the claimed invention, as the invention must be considered as a whole. *In re Hirao*, 535 F.2d 67, 190 U.S.P.Q. 15 (C.C.P.A. 1976).

The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, not in Appellant's disclosure. *In re Vaeck*, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991).

No Motivation to Combine

In the present case, none of these criteria have been met in the Final Office Action. First, there is no suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the BLAST method of Altschul et al. or combine it with Fujimiya et al., Anderson et al., Lincoln et al., and Matsumoto et al.

The BLAST disclosure of Altschul et al. is concerned with rapid sequence comparison for sequence database searches (i.e., a sequence compared to a database), motif searches (again, a sequence compared to a database or library of motifs), gene identification searches (another sequence comparison to a database), and analysis of regions of similarity in long DNA sequences (a sequence compared to a sequence). Altschul et al. admits at page 405, lines 19-26 that disk space requirements are high and random access to the database are slow even for query sequences of *typical length*, making scanning the entire database [into RAM] a faster method (see, e.g., Appellants' "Prior art" figure 1 for an example of this faster method), *which, as a whole, teaches away from the present invention.*

Likewise, Fujimiya et al. teaches the use of a single CPU that accesses the entire target database and key database (i.e., CD-ROM) and fails to teach or fairly suggest use of a *distributed* computing platform, but instead *teaches away from the present invention by teaching serial processing* (see, e.g., "by transmitting the sequence data of the bases from the gene database continually one after another into the dynamic programming operation unit as target data" in the Abstract). Contrary to the assertions at page 6 of Paper No. 4, the disclosed M and N in Fujimiya et al. merely refers to the size of sequence data of the bases that are fed at a time to the dynamic

programming operation unit. The "data element" bases of M and N are inherently unitary and therefore cannot be "a size within a specified range" as claimed. The Examiner has merely taken the similar language and notation out of context and twisted it to appear to match Appellants' claims.

Anderson et al. discloses a study of sequence database searching algorithms to determine sensitivity and coverage of the algorithms so as to form guidelines to be used in accessing the significance of DNA database search results obtained by the algorithms (see, e.g., the Abstract). Various-sized base pair sequences were searched against various databases (see pages 351-352 and tables 1 and 2). Contrary to the assertion at pages 6-7 of Paper No. 4, Anderson et al. does not teach or suggest the claimed task definitions. The Examiner has again cited portions of the prior art related to results that have similar language in an attempt to mischaracterize the prior art disclosure to match Appellants' claims.

The mischaracterization continues in the Examiner's discussion of Lincoln et al., which essentially discloses a relational database that allows analysis both within the relational database and between the information in the relational database and other public databases. It has nothing to do with any distributed computing platform and the Examiner's overly broad interpretation of the claimed master-slave CPU distributed computing platform is not a "reasonable interpretation consistent

with the specification" as required by M.P.E.P. 2111.

Matsumoto et al. discloses various biological sequence data compression algorithms, but fails to teach or fairly suggest the claimed distributed computing platform.

With respect to a suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify Altschul et al. or to combine the cited reference teachings, pages 8-9 of Paper No. 4 cites portions of the modifying references that allegedly suggest the need for: i) significant computational resources, ii) practical processing time, iii) efficient sequence data storage and communication, and iv) efficient computerized sequence database analysis and comparison.

In other words, the prior art was aware of various needs and problems with respect to biological sequence comparison. However, many of the cited prior art references disclose solutions that, as a whole, *teach against* the presently claimed invention, as discussed above. The Final Rejection does not even state what elements of Altschul et al. would have been obvious to modify - but rather that it would have been obvious to obtain a result - to improve speed and accuracy. Furthermore, despite allegedly recognizing the problems/needs of the art, none of the cited references even teaches or fairly suggests use of a distributed computing platform for sequence data comparison, let alone the specific implementation claimed by Appellants.

Indeed, the present invention allows large dataset (database) to dataset (database) comparisons by the segmentation of the datasets and the subsequent segmentation of the comparison tasks for performance of the tasks on a distributed computing platform. The claimed methods and means have numerous advantages over the prior art. It allow tools for sequence to sequence comparison, such as BLAST, to be used to compare very large datasets (such as databases). It lets these computations be performed on modest computational hardware, it reduces the time involved by using parallel processing, and it leverages the unused computational cycles of desktop computers.

Clearly, the stated motivation to combine is wrought with errors and hindsight. On page 8 of the Paper No. 4, it says: "A skilled artisan in the art would have been motivated to make improvements to a rapid homology retrieval program [Altschul et al.]...on various datasets in order to provide faster and more accurate access of the information to users [Fujimiya et al. and Matsumoto et al.]" And as previously submitted, the present invention is not to "faster and more accurate homology retrieval," but to comparisons of larger datasets with multiple CPUs having modest RAM through use of a distributed computing platform.

Pages 8-9 continue with: "Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to test datasets [relying on Anderson et al.,

which teaches against the present invention by teaching ordinary sequence to database comparison] ...containing possible combinations [relying on Fujimiya et al., which teaches against the present invention by teaching serial processing at a single site]...using...BLAST [Altschul et al.]..., and speeding up the process using compressed files [Matsumoto et al.]...and looping the sequences analyzed [Lincoln et al.] ...with generated reports which could all be sent over a network of CPUs and databases in order to allow greater, faster, and efficient access to users of the homology information via methods and a computer system [relying on Fujimiya et al. which teaches against a distributed computing platform and Matsumoto et al. which merely teaches data compression].

However, as previously submitted, the claimed invention does not concern and has no limitations to "test datasets" as in Anderson et al., but rather divides a comparison of datasets N and M into $n_N \times n_M$ tasks; the claimed invention does not speed "up the process using compressed files," but rather uses efficient data structures and compression to lower the use of network bandwidth; the claimed invention has limitations to "looping the sequences analyzed" in claim 18, but also requires "looping through query sequences from said query data element to perform setup, preprocessing and table generation for each row of comparisons" that is not suggested; the present invention is not concerned with generating "reports which could all be sent over a

network of CPUs and databases in order to allow greater, faster, and efficient access to users of the homology information via methods and a computer system," but rather is drawn to tasking multiple slave CPUs with comparing a portion of dataset N with a portion of dataset M that the slave CPU has resources to perform within a given time; the "reports" are not sent to users, but are sent to the master CPU for concatenating into a text file identical to a *blastall* run.

As most of these "motivational" statements have little to do with the present invention, it is clear that the references were cobbled together in hindsight in a feeble attempt to find literature that includes language similar to the claimed limitations.

No Reasonable Expectation of Success

One of ordinary skill in the art could not reasonably be expected to find Appellants' claimed invention for comparing large datasets obvious in view of a plurality of references that provide no guidance on handling large datasets or processing them in parallel over a network using a distributed computing platform.

All Claim Limitations Not Shown

Altschul et al. disclose the basic BLAST algorithm for sequence comparison, i.e., comparing one sequence with another sequence, or for searching a database. However, Altschul et al.

at least fail to disclose or suggest dividing sequence comparison problems into discrete segments for processing on a plurality of CPUs, let alone any specific method of doing this task.

Fujimiya et al. disclose a dynamic programming method for sequence comparison for searching a database. However, like Altschul et al., Fujimiya et al. at least fail to disclose or suggest dividing sequence comparison problems into discrete segments for processing on a plurality of CPUs, let alone any specific method of doing this task.

Anderson et al. disclose a study into finding significant matches when matching DNA sequences to sequence databases, using BLAST, BLAST2, FASTA, etc. The paper does not disclose or suggest dividing sequence comparison problems into discrete segments for processing on a plurality of CPUs, let alone any specific method of doing this task.

Lincoln et al. disclose a database for storage and analysis of full-length genetic sequences. This patent does not disclose or suggest dividing sequence comparison problems into discrete segments for processing on a plurality of CPUs, let alone any specific method of doing this task.

Further, the Examiner's interpretation of the database and external public databases as "master and slave CPUs" is entirely erroneous. The "broadest reasonable interpretation" of a claim element under M.P.E.P. §2111 must be an interpretation of the claim element, not the prior art, and must also be "consistent

APPELLANT'S BRIEF ON APPEAL
U.S. Application No. 09/881,234

with the specification." In the present case, the Office Action has failed to explain in what possible manner "a method and system of storing and retrieving data from a database and external public databases" could possibly read on the claimed master and slave CPUs as claimed

in claim 1:

"sending all data elements and task definitions to a master central processing unit (CPU) of a master-slave distributed computing platform,

wherein task definitions comprise at least one comparison parameter, at least one executable element capable of performing comparisons, a query data element identification(ID)/descriptor, and a subject data element ID/descriptor, and

wherein data elements are sent alternately from query and subject data elements;

sending a task definition for each task from the master CPU to one of a plurality of slave CPUs when all parts of a task definition and data elements referenced by said task definition are available at said master CPU;

sending data elements referenced by said task definition to said slave CPU;

performing each task on a slave CPU; and

returning task results for each task to said master CPU"

or in claim 13:

"a master central processing unit (CPU) of a master-slave distributed computing platform;

a plurality of slave CPUs capable of communication with said master CPU; and

a client CPU with instructions for:

dividing said query dataset N into n_N data elements having a size within a specified range;

dividing said subject dataset M into n_M data elements having a size within said specified range;

determining a number of tasks for an entire comparison of datasets N and M as $n_N \times n_M$;

sending all data elements and task definitions to said master CPU of a master-slave distributed computing platform, wherein task definitions comprise at least one

comparison parameter, at least one executable element capable of performing comparisons, a query data element identification(ID)/descriptor, and a subject data element ID/descriptor, and wherein data elements are sent alternately from query and subject data elements;
said master CPU comprising instructions for:
 sending a task definition for each task to one of said plurality of slave CPUs when all parts of a task definition and data elements referenced by said task definition are available at said master CPU; and
 sending data elements referenced by said task definition to said slave CPU; and
said slave CPUs including instructions for:
 performing each task; and
returning task results for each task to said master CPU."

Indeed, none of the cited prior art discloses or fairly suggests a master CPU/plurality of slave CPU's for dividing and processing a unitary sequence comparison.

Matsumoto et al. disclose biological sequence compression algorithms, but like all of the other cited prior art, fail to disclose or suggest dividing sequence comparison problems into discrete segments for processing on a plurality of CPUs, let alone any specific method of doing this task.

One reason that none of this prior art discloses or fairly suggests dividing sequence comparison problems into discrete segments for processing on a plurality of CPUs, let alone any specific method of doing this task, is because *they are not intended for comparing one database to another*, but rather for comparing one sequence to another or to a database. The prior art way of comparing a database to a database is to divide one database into sequences and then run the basic sequence-to-

database comparison multiple times, possibly in parallel on multiple CPUs, as discussed by Appellants with respect to Prior Art Figure 1A. The present invention splits the problem of comparing datasets M and N into $n_N \times n_M$ comparisons of data elements from N with data elements from M, as illustrated, for comparison with Figure 1A, in Figure 1B.

As a whole, none of the cited prior art teaches or fairly suggests dividing the problem of comparing datasets M and N into $n_N \times n_M$ comparisons of data elements from N with data elements from M as presently claimed. For at least these reasons, Appellants submit that the claims are allowable over the prior art and requests reconsideration and allowance of the claims.

Reply to the Examiner's Response to Arguments

Appellants submit that, although the Examiner is correct in stating that the motivation to combine references need not be the same motivation as Appellants', Paper No. 4's long string of hazy motivational statements is evidence of an improper attempt at hindsight reconstruction of the claimed invention.

With respect to the discussion of no reasonable expectation of success, the Examiner improperly argues the *separate requirement* of motivation (which also fails for the reasons stated above).

With respect to whether all the claim limitations are taught

APPELLANT'S BRIEF ON APPEAL
U.S. Application No. 09/881,234

or suggested, the Examiner, by looking only to lines 2-5 of claim 1, improperly ignores numerous claim limitations that are related to "dividing sequence comparison problems into discrete segments *for processing on a plurality of CPUs, let alone any specific method of doing this task*" as argued by Appellants -- i.e., using a distributed computing platform, as argued on pages 10-11 of the response filed July 14, 2003. Indeed, the remainder of the Examiner's arguments with respect to Altschul et al. are related to selection of the sequence size, not to further subdivision of both the query and subject (target) datasets or the use of a distributed computing platform involving numerous other claimed steps or means. Likewise, Anderson et al. discloses the selection (not segmentation) of *query* sequences of various base-pair sizes, but nowhere suggests *segmentation* of either the query or subject (target) datasets. As with Altschul et al., the Examiner ignores Appellants' arguments related to the distributed computing platform, which involves numerous other claimed steps or means.

In a similar manner, the Examiner's arguments concerning the claimed slave-master CPU distributed computing platform completely ignore basic elements of the Appellants' position. First, the terms "master CPU," "slave CPU," and "master-slave distributed computing platform" have established meanings to one of skill in the art. A master CPU controls and directs the actions (tasks) of the slave CPUs. The relational database and

other public databases in Lincoln et al. do not meet this established definition for the terms.

Additionally, Appellants' specification discusses prior art systems, including use of a central database, GenBank, in paragraph [4], and of programs such as SSEARCH, FASTA, and BLAST that conform to the one query sequence, one database paradigm in paragraph [8]. Researcher computer access to GenBank is inherent and constitutes a server-client paradigm that the Examiner apparently believes is reasonably included in the claimed master-slave CPU arrangement. However, the disclosed and claimed master-slave CPU arrangement of the present invention is clearly distinguished from the disclosed and inherent prior art in the specification, such that the Examiner's "broadest reasonable interpretation" is clearly not "consistent with the specification."

The specification even alludes to previous attempts at distributed computing at paragraph 13, but nowhere describes this system as falling within a *master-slave CPU* distributed computing platform:

"In the naive method of dividing a database-to-database comparison into multiple subcomparisons, the query database is divided into multiple smaller query sub-databases. Each query sub-database is sent to a separate CPU, as well as the entire subject database, and the comparison is performed. As noted above, this method requires large amounts of RAM to efficiently compare queries to a large subject database. In addition, it also means that for each new CPU employed in the search, the amount of data that must be transferred to that CPU includes the entire subject database. As the number of CPUs employed in the search increases, the total

amount of data transferred when initiating a large database-to-database comparison is effectively the number of CPUs times the size of the subject database. This typically saturates the network on which the machines reside, diminishing the usefulness of each new CPU added to the network."

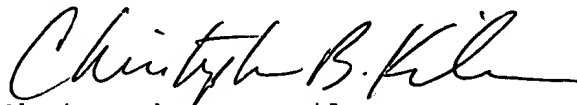
This type of distributed computing is illustrated in figure 1A and, despite being vastly more similar to the claimed invention than Lincoln et al., still does not satisfy the claim limitations. The Examiner's position is clearly unreasonable.

IX. CONCLUSION

For the above reasons, Appellants respectfully submit that the present claims meet the requirements of 35 U.S.C. 112 and that the Examiner has failed to make out a *prima facie* case of obviousness under 35 U.S.C. 103 with regard to claims 1, 4, 6-7, 9-10, 12-13, 16, 18-21 and 23 and asks that the obviousness rejection be reversed.

The present Brief on Appeal is being filed in triplicate.

Respectfully submitted,



Christopher B. Kilner
Registration No. 45,381
Roberts Abokhair & Mardula, LLC
11800 Sunrise Valley Drive, Suite 1000
Reston, VA 20191
(703) 391-2900

APPENDIX

- [c1] A method of comparing a query dataset N with a subject dataset M, comprising:
- dividing said query dataset N into n_N data elements having a size within a specified range;
- dividing said subject dataset M into n_M data elements having a size within said specified range;
- determining a number of tasks for an entire comparison of datasets N and M as $n_N \times n_M$;
- sending all data elements and task definitions to a master central processing unit (CPU) of a master-slave distributed computing platform,
- wherein task definitions comprise at least one comparison parameter, at least one executable element capable of performing comparisons, a query data element identification(ID)/descriptor, and a subject data element ID/descriptor, and
- wherein data elements are sent alternately from query and subject data elements;
- sending a task definition for each task from the master CPU to one of a plurality of slave CPUs when all parts of a task definition and data elements referenced by said task definition are available at said master CPU;
- sending data elements referenced by said task definition to said slave CPU;
- performing each task on a slave CPU; and
- returning task results for each task to said master CPU.
- [c2] The method of claim [c1], further comprising randomizing sequence order of each dataset if either dataset contains related sequences in a contiguous arrangement.
- [c3] The method of claim [c1], further comprising formatting said datasets so as to use exactly

the same ambiguity substitutions.

- [c4] The method of claim [c1] wherein dividing said datasets into data elements further comprises:

stripping all metadata from data;

packing said data into an efficient structure;

creating an index for said data and packing said index and said data in an uncompressed data structure; and

compressing said uncompressed data structure into a data element using a redundancy reduction data compression method.

- [c5] The method of claim [c1], further comprising sending remaining data elements from a more numerous of said datasets to said master CPU followed by all task definitions for otherwise complete tasks if there are fewer data elements from one dataset.

- [c6] The method of claim [c1] wherein performing a task on said slave CPU further comprises:

uncompressing and unpacking data from said query and subject data elements;

looping through query sequences from said query data element to perform setup, preprocessing and table generation for each row of comparisons;

looping through subject sequences from said subject data element and, for each pair of query and subject sequences, performing a comparison using said executable element and finding results based on said at least one comparison parameter; and

storing minimal information that will allow reconstruction of said result.

- [c7] The method of claim [c6] wherein storing said minimal information comprises:

storing index information for said query and said subject sequence;

storing bounds information for start and stop of said query and subject sub sequences;

storing data that quantify fulfillment of significance criteria for a significant match; and

storing an efficiently encoded representation of alignment between said bounds corresponding to a high-scoring segment pair.

[c8] The method of claim [c7], further comprising storing a seed point and sum-set membership for each alignment for Basic Local Alignment Search Tool (BLAST).

[c9] The method of claim [c7], further comprising storing task results in a task result file, said file including query and subject sequence data and metadata corresponding to the task that the results came from, metadata for the subject sequence, the partial subject sequence data corresponding to the subject bounds of the significant alignment result, and any other results data for each result in the task results.

[c10] The method of claim [c9], further comprising generating a BLAST report for each query data element.

[c11] The method of claim [c10], further comprising concatenating results from all BLAST reports to produce a text file identical to a blastall run of said query and subject datasets.

[c12] The method of claim [c1] wherein said datasets are selected from the group consisting of genomic and proteomic databases.

[c13] A system for comparing a query dataset N with a subject dataset M, comprising:

a master CPU of a master-slave distributed computing platform;

a plurality of slave CPUs capable of communication with said master CPU; and

a client CPU with instructions for:

dividing said query dataset N into n_N data elements having a size within a specified range;

dividing said subject dataset M into n_M data elements having a size within said specified range;

determining a number of tasks for an entire comparison of datasets N and M as $n_N \times n_M$;

sending all data elements and task definitions to said master central processing unit (CPU) of a master-slave distributed computing platform,

wherein task definitions comprise at least one comparison parameter, at least one executable element capable of performing comparisons, a query data element identification(ID)/descriptor, and a subject data element ID/descriptor, and

wherein data elements are sent alternately from query and subject data elements;

said master CPU comprising instructions for:

sending a task definition for each task to one of said plurality of slave CPUs when all parts of a task definition and data elements referenced by said task definition are available at said master CPU; and

sending data elements referenced by said task definition to said slave CPU; and

said slave CPUs including instructions for:

performing each task; and

returning task results for each task to said master CPU.

[c14] The system of claim [c13], further comprising means for randomizing sequence order of each dataset if either dataset contains related sequences in a contiguous arrangement.

[c15] The system of claim [c13], further comprising means for formatting said datasets so as to use exactly the same ambiguity substitutions.

[c16] The system of claim [c13], wherein said instructions for dividing said datasets into data elements further comprises instructions for:

stripping all metadata from data;

packing said data into an efficient structure;

creating an index for said data and packing said index and said data in an uncompressed data structure; and

compressing said uncompressed data structure into a data element using a redundancy reduction data compression method.

[c17] The system of claim [c13], further comprising instructions for sending remaining data elements from a more numerous of said datasets to said master CPU followed by all task definitions for otherwise complete tasks if there are fewer data elements from one dataset.

[c18] The system of claim [c13], wherein instructions for performing a task on said slave CPU further comprises instructions for:

uncompressing and unpacking data from said query and subject data elements;

looping through query sequences from said query data element to perform setup, preprocessing and table generation for each row of comparisons;

looping through subject sequences from said subject data element and, for each pair of query and subject sequences, performing a comparison using said executable element and finding results based on said at least one comparison parameter; and

storing minimal information that will allow reconstruction of said result.

[c19] The system of claim [c18], wherein said instructions for storing said minimal information

comprises instructions for:

storing index information for said query and said subject sequence;

storing bounds information for start and stop of said query and subject sub sequences;

storing data that quantify fulfillment of significance criteria for a significant match; and

storing an efficiently encoded representation of alignment between said bounds
corresponding to a high-scoring segment pair.

[c20] The system of claim [c19], further comprising instructions for storing task results in a task result file, said file including query and subject sequence data and metadata corresponding to the task that the results came from, metadata for the subject sequence, the partial subject sequence data corresponding to the subject bounds of the significant alignment result, and any other results data for each result in the task results.

[c21] The system of claim [c20], further comprising instructions for generating a Basic Local Alignment Search Tool (BLAST) report for each query data element.

[c22] The system of claim [c21], further comprising means for concatenating results from all BLAST reports to produce a text file identical to a blastall run of said query and subject datasets.

[c23] The system of claim [c13], wherein said datasets are selected from the group consisting of genomic and proteomic databases.

APPELLANT'S BRIEF ON APPEAL
U.S. Application No. 09/881,234

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of

Atty. Docket No.: 2551-026

Douglas M. BLAIR

Appln. No.: 09/881,234

Group Art Unit: 1631

Filing Date: June 14, 2001

Examiner: Smith, C.

For: APPARATUS AND METHOD FOR PROVIDING SEQUENCE DATABASE
COMPARISON

* * * * *

APPELLANT'S AMENDMENT UNDER 37 C.F.R. § 1.116

* * * * *

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

In accordance with the provisions of 37 C.F.R. § 1.116, Appellant submits the following amendment to place the claims in better condition for appeal. This amendment materially reduces the issues on appeal and should therefore be entered.

Amendments to the Specification begin on page 38;

Amendments to the Claims begin on page 39.

AMENDMENT

In the Specification:

Please amend the specification as follows:

- [11] While these large-scale computational resources have made many of the contemporary advances in genomic research possible, there are significant shortcomings to this approach. The NCBI BLAST (Basic Local Alignment Search Tool) program *blastall* is probably the most extensively used program in the suite of BLAST programs. It performs the *blastn* (nucleotide query sequence vs. nucleotide subject database), *blastp* (amino acid sequence vs. protein subject database), *blastx* (translated nucleotide sequence vs. protein subject database), *tblastn* (amino acid sequence vs. translated nucleotide subject database), and *tblastx* (nucleotide query sequence vs. nucleotide subject database, both translated) algorithms. In comparing one query sequence at a time to a subject database, *blastall* must scan through all the data in the subject database for each query sequence. The basic BLAST algorithm is fast enough that it spends very little CPU (central processing unit) time with each subject sequence. As a result, the speed with which the algorithm can process subject data significantly exceeds the speed of even the fastest local disks, and far exceeds the speed of most network accessible disks, which are often where large datasets reside.

AMENDMENT

In the Claims:

Please amend the claims as follows:

[c1] (Currently amended) A method of comparing a query dataset N with a subject dataset M, comprising:

dividing said query dataset N into n_N data elements having a size within a specified range;

dividing said subject dataset M into n_M data elements having a size within said specified range;

determining a number of tasks for an entire comparison of datasets N and M as $n_N \times n_M$;

sending all data elements and task definitions to a master central processing unit (CPU) of a master-slave distributed computing platform,

wherein task definitions comprise at least one comparison parameter, at least one executable element capable of performing comparisons, a query data element identification(ID)/descriptor, and a subject data element ID/descriptor, and

wherein data elements are sent alternately from query and subject data elements;

sending a task definition for each task from the master CPU to one of a plurality of slave CPUs when all parts of a task definition and data elements referenced by said task definition are available at said master CPU;

sending data elements referenced by said task definition to said slave CPU;

performing each task on a slave CPU; and

returning task results for each task to said master CPU.

[c2] (Original) The method of claim [c1], further comprising randomizing sequence order of each dataset if either dataset contains related sequences in a contiguous arrangement.

[c3] (Original) The method of claim [c1], further comprising formatting said datasets so as to use exactly the same ambiguity substitutions.

[c4] (Original) The method of claim [c1] wherein dividing said datasets into data elements further comprises:

stripping all metadata from data;

packing said data into an efficient structure;

creating an index for said data and packing said index and said data in an uncompressed data structure; and

compressing said uncompressed data structure into a data element using a redundancy reduction data compression method.

[c5] (Original) The method of claim [c1], further comprising sending remaining data elements from a more numerous of said datasets to said master CPU followed by all task definitions for otherwise complete tasks if there are fewer data elements from one dataset.

[c6] (Original) The method of claim [c1] wherein performing a task on said slave CPU further comprises:

uncompressing and unpacking data from said query and subject data elements;

looping through query sequences from said query data element to perform setup, preprocessing and table generation for each row of comparisons;

looping through subject sequences from said subject data element and, for each pair of query and subject sequences, performing a comparison using said executable element and finding results based on said at least one comparison parameter; and

storing minimal information that will allow reconstruction of said result.

[c7] (Original) The method of claim [c6] wherein storing said minimal information comprises:

storing index information for said query and said subject sequence;

storing bounds information for start and stop of said query and subject sub sequences;

storing data that quantify fulfillment of significance criteria for a significant match; and

storing an efficiently encoded representation of alignment between said bounds corresponding to a high-scoring segment pair.

[c8] (Currently amended) The method of claim [c7], further comprising storing a seed point and sum-set membership for each alignment for Basic Local Alignment Search Tool (BLAST).

[c9] (Original) The method of claim [c7], further comprising storing task results in a task result file, said file including query and subject sequence data and metadata corresponding to the task that the results came from, metadata for the subject sequence, the partial subject sequence data corresponding to the subject bounds of the significant alignment result, and any other results data for each result in the task results.

[c10] (Original) The method of claim [c9], further comprising generating a BLAST report for each query data element.

[c11] (Original) The method of claim [c10], further comprising concatenating results from all BLAST reports to produce a text file identical to a blastall run of said query and subject datasets.

[c12] (Original) The method of claim [c1] wherein said datasets are selected from the group consisting of genomic and proteomic databases.

[c13] (Currently amended) A system for comparing a query dataset N with a subject dataset M, comprising:

a master central processing unit (CPU) of a master-slave distributed computing platform;

a plurality of slave CPUs capable of communication with said master CPU; and

a client CPU with instructions for:

dividing said query dataset N into n_N data elements having a size within a specified range;

dividing said subject dataset M into n_M data elements having a size within said specified range;

determining a number of tasks for an entire comparison of datasets N and M as $n_N \times n_M$;

sending all data elements and task definitions to said master CPU of a master-slave distributed computing platform,

wherein task definitions comprise at least one comparison parameter, at least one executable element capable of performing comparisons, a query data element identification (ID)/descriptor, and a subject data element ID/descriptor, and

wherein data elements are sent alternately from query and subject data elements;

said master CPU comprising instructions for:

sending a task definition for each task to one of said plurality of slave CPUs when all parts of a task definition and data elements referenced by said task definition are available at said master CPU; and

sending data elements referenced by said task definition to said slave CPU; and

said slave CPUs including instructions for:

performing each task; and

returning task results for each task to said master CPU.

[c14] (Original) The system of claim [c13], further comprising means for randomizing sequence order of each dataset if either dataset contains related sequences in a contiguous arrangement.

[c15] (Original) The system of claim [c13], further comprising means for formatting said datasets so as to use exactly the same ambiguity substitutions.

[c16] (Original) The system of claim [c13], wherein said instructions for dividing said datasets into data elements further comprises instructions for:

stripping all metadata from data;

packing said data into an efficient structure;

creating an index for said data and packing said index and said data in an uncompressed data structure; and

compressing said uncompressed data structure into a data element using a redundancy reduction data compression method.

[c17] (Original) The system of claim [c13], further comprising instructions for sending remaining data elements from a more numerous of said datasets to said master CPU followed by all task definitions for otherwise complete tasks if there are fewer data elements from one dataset.

[c18] (Original) The system of claim [c13], wherein instructions for performing a task on said slave CPU further comprises instructions for:

uncompressing and unpacking data from said query and subject data elements;

looping through query sequences from said query data element to perform setup, preprocessing and table generation for each row of comparisons;

looping through subject sequences from said subject data element and, for each pair of

query and subject sequences, performing a comparison using said executable element and finding results based on said at least one comparison parameter; and

storing minimal information that will allow reconstruction of said result.

[c19] (Original) The system of claim [c18], wherein said instructions for storing said minimal information comprises instructions for:

storing index information for said query and said subject sequence;

storing bounds information for start and stop of said query and subject sub sequences;

storing data that quantify fulfillment of significance criteria for a significant match; and

storing an efficiently encoded representation of alignment between said bounds corresponding to a high-scoring segment pair.

[c20] (Original) The system of claim [c19], further comprising instructions for storing task results in a task result file, said file including query and subject sequence data and metadata corresponding to the task that the results came from, metadata for the subject sequence, the partial subject sequence data corresponding to the subject bounds of the significant alignment result, and any other results data for each result in the task results.

[c21] (Currently amended) The system of claim [c20], further comprising instructions for generating a Basic Local Alignment Search Tool (BLAST) report for each query data element.

[c22] (Original) The system of claim [c21], further comprising means for concatenating results from all BLAST reports to produce a text file identical to a blastall run of said query and subject datasets.

[c23] (Original) The system of claim [c13], wherein said datasets are selected from the group consisting of genomic and proteomic databases.

Remarks

The present amendment materially reduces the issues on appeal by addressing the 112 rejections related to the use of abbreviations in the claims.

Respectfully submitted,

A handwritten signature in black ink, appearing to read "Christopher B. Kilner". The signature is fluid and cursive, with a long horizontal stroke at the end.

Christopher B. Kilner
Registration No. 45,381
Roberts Abokhair & Mardula, LLC
11800 Sunrise Valley Drive
Suite 1000
Reston, VA 20191
(703) 391-2900